The project is an encrypted chat client that uses RSA to encrypt messages across a network.

To begin, the server sets up a server and begins accepting clients. Clients, upon acceptance are handled in individual threads. Clients are accepted with 2 socket descriptors for commands and for mail.

Once accepted the client, if new, must register a new user.
Then, regardless of first time connection, users proceed to the login phase:

The client and server perform an RSA handshake, read on for details:

> During the login phase, the server generates a random token, encrypts it with the client's public key, and sends the encrypted message.
>
> The client then decrypts the received token with its private and public key, then encrypts the token with the server's public key and sends it to the server
>
> The server finally decrypts the received token and verifies that the token is correct. If so, it logs in the client and saves its mail socket address in a hashmap.

After logging in, the client can call the /listonline function to see who is online. It can use /msg to Talk to individual users or /mall to talk to everyone.

Upon disconnecting, the server will log out the client.

Specifics:

KNOWN ERRORS:

> When running the server, it will fail to bind to the same port again sometimes. This can be fixed by changing the port arg

HOW TO COMPILE AND RUN:

```
cd ~
wget https://ftp.gnu.org/gnu/gmp/gmp-5.1.3.tar.xz
tar xf gmp-5.1.3.tar.xz
cd gmp-5.1.3
./configure --prefix=$HOME
make
make install
```

Cd to the program dir
Compile the program with make

RUNNING THE PROGRAM:
Run each client/the server from different folders because the clients/the server will read the key stored from where the program is run.

So to run 2 clients and a server:

mkdir folder1
mkdir folder2
mkdir folder3
mkdir server1
cp client folder1/client
cp client folder2/client
cp client folder3/client
cp server server1/server
From folder1: ./client -p [port] -h [host]
From folder2: ./client ----
From folder3: ./client ----
From server1: ./server -p [port]

OR
Just make the 4 folders and run
From folderN ../client ----
From server1 ../server ---

The server and the client cannot be run from the same directory

The makefile places the executables in different folders for convenience
To make multiple instances of clients just make copies of the created client folder

COMMANDS INVOLVING THE SERVER:

Register:
The client begins by sending data with the information to register itself on the server
The server then sends an ACK/NACK telling it whether it really worked

Login:
The client begins by sending a login request with its username.
The server responds by sending back the encrypted token.
The client responds by decrypting the token and encrypting the token with the serverkey.
Then its sending back the newly encrypted token.
Each of these processes are encapsulated by acks to prevent errors.

Upon succeeding the server then stores the client's mail socket address in a hashmap so it can receive messages.

Messaging:

The client requests the public key of the recipient from the server;
It then encrypts the metadata [destination, recipient,ect] with the server public key and encrypts the message with the recipient’s public key.
It sends this message to the server. The server can only read the metadata and not the message itself because the message is in the final recipients public key.
The message metadata is decrypted,read, and encrypted with the recipient's public key
The message is written the the correct recipients mail socket as retrieved from the hashmap filled during login.

Message All:

Same as message but does it for each client available

List all:

Server sends the client a string of text containing all online users

Exit:

Client tells the server it wants to leave and then the server removes it